

УДК 519.632.4

ФОРМАЛИЗАЦИЯ ВЫЧИСЛЕНИЙ ПРИ ЧИСЛЕННОМ РЕШЕНИИ КРАЕВЫХ ЗАДАЧ

С.И. Мартыненко

Аннотация

Рассматривается один из подходов к решению проблемы формализации вычислений при численном решении широкого класса краевых задач со скоростью сходимости, близкой к оптимальной. Дано описание основных компонент универсальной многосеточной технологии и результатов вычислительных экспериментов.

Ключевые слова: краевая задача, многосеточные методы, метод конечных элементов, вычислительный эксперимент.

Введение

Начиная с середины 1980-х годов габариты и стоимость вычислительной техники стали стремительно уменьшаться, а производительность стала стремительно нарастать. К компьютерам получили массовый доступ инженеры, физики, химики и другие специалисты, которые не имели достаточной подготовки в области вычислительной математики, но у которых были свои задачи, зачастую связанные с необходимостью численного решения дифференциальных уравнений в частных производных. Сразу возникла проблема эффективного использования компьютерной техники пользователями, которые не обладают достаточными навыками программирования и познаниями в области численных методов.

Выход был найден в разработке *автономных*¹ программ, используя которые инженер только формулирует задачу, а детали вычислительного алгоритма ему могут быть даже неизвестны. Применительно к техническим приложениям, работу автономных программ упрощенно можно представить следующим образом: конструктор проектирует некоторый узел при помощи графической программы, например AutoCAD. Затем геометрия узла переносится в вычислительный модуль, конструктор задает граничные и начальные условия, после чего проводит тепловой, прочностной, гидродинамический или другой расчет и анализирует результаты. Как правило, после анализа полученных результатов нужно внести изменения в конструкцию и повторить расчет. Обычно конструктор выполняет несколько подобных «итераций», чтобы получить оптимальную, с его точки зрения, конструкцию. Еще большую практическую ценность будет представлять возможность расчета машины в целом, а не только отдельных ее узлов, поскольку поэлементное моделирование часто связано с погрешностями постановки граничных условий.

Однако широкому внедрению методологии математического моделирования в машиностроение препятствует несовершенство численных методов. Применение традиционных численных методов подразумевает возможность контроля математиком хода вычислительного процесса и внесения изменений в написанную им же

¹В англоязычной литературе, помимо термина *autonomous software*, иногда используется термин *black box software*, то есть программы, устроенные по принципу «черного ящика» [1].

программу. Ранее такой подход был оправдан, но с появлением мощных персональных компьютеров возникла острая потребность в новых вычислительных методах, обладающих достаточной точностью, эффективностью, высоким уровнем формализации и параллелизма без контроля со стороны пользователя. Уже сейчас такие программные продукты, как ANSYS, STAR-CD, FLUENT, CFX, PHOENICS и др., получили широкое применение в НИИ и ОКБ для решения разнообразных прикладных задач. Конечно, подобные программные средства еще далеки от совершенства и часто вызывают обоснованные претензии пользователей, что обусловлено использованием в них традиционных численных методов. Однако на автономные продукты существует устойчивый спрос, а значит, будет и предложение. Конкуренция на рынке программного обеспечения неизбежно приведет к повышению их потребительских качеств.

Сформулируем задачу о построении *универсального алгоритма* для автономных программных продуктов. Пусть дана исходная дифференциальная задача

$$\mathcal{L}(u) = f(x),$$

где $x \in \Omega$, Ω – область N -мерного пространства, $f(x)$ – заданная функция, \mathcal{L} – (не)линейный дифференциальный оператор. Полагается, что граничные условия учтены оператором \mathcal{L} и правой частью $f(x)$. В области Ω строится сетка и/или триангуляция, на которой осуществляется контрольно-объемная и/или конечно-элементная аппроксимация дифференциальной задачи. Обозначим результирующую СЛАУ, полученную в результате аппроксимации, как

$$Ax = b.$$

Предположим, что имеется множество итерационных методов \mathcal{M} , при помощи которых можно решить полученную СЛАУ. Тогда задача о построении универсального алгоритма состоит в отыскании такой стратегии применения итерационных методов из множества \mathcal{M} к решению СЛАУ $Ax = b$, которая позволит достичь скорости сходимости, близкой к оптимальной, то есть для численного решения краевых задач нужно выполнить $O(\bar{N} \lg^k \bar{N})$ арифметических операций, где \bar{N} есть число неизвестных, а $k = 1, 2$.

При этом на универсальный алгоритм накладываются следующие ограничения:

1. *Минимальное использование ресурсов компьютера.* Темпы развития современных математических моделей опережают совершенствование аппаратной части вычислительной техники. Поэтому в отдельных случаях ресурсы компьютера могут быть задействованы преимущественно для хранения разностной задачи, а не промежуточных вычислений. Минимальный объем хранимых данных составит $\sim 2\bar{N}$ вещественных чисел, то есть в оперативной памяти могут размещаться только вектора x и b результирующей СЛАУ. В общем случае матрицу коэффициентов A не имеет смысла хранить, поскольку ее элементы будут пересчитываться после каждой итерации по нелинейности. Рациональное использование ресурсов компьютера позволит применять достаточно мелкие вычислительные сетки для численного решения краевых задач в автономных программах.

2. *Возможность изменения способа и/или порядка аппроксимации в процессе решения задачи.*

3. *Отсутствие проблемно-зависимых компонент.* В универсальном алгоритме не допускается применение компонент, которые зависят от решаемой задачи и обеспечивают высокую скорость сходимости. Однако возможно использование проблемно-зависимых параметров (например, верхней и/или нижней релаксации) при условии, что существует эффективная методика определения их оптимальных значений без контроля со стороны пользователя.

4. *Возможность эффективного распараллеливания вычислений вне зависимости от используемых итерационных методов из множества \mathcal{M} .*

По-видимому, отсутствие проблемно-зависимых компонент в универсальном алгоритме не позволит достичь оптимальной скорости сходимости ($k = 0$), хотя этот факт не доказан.

Очевидна невозможность применения только одного итерационного метода в автономных программах, поэтому нужно научиться комбинировать различные алгоритмы для получения универсальной и высокоэффективной вычислительной технологии. Для этого понадобилась принципиально новая идея, которую предложил выдающийся отечественный математик Радий Петрович Федоренко. В 1961 и 1964 гг. им были опубликованы две работы, в которых использовалась высокая сходимость некоторых итерационных методов для высокочастотных гармоник [2, 3]. В 1966 г. Н.С. Бахвалов доказал оптимальность метода по числу арифметических операций для достижения точности, согласованной с порядком аппроксимации [4]. Многосеточные методы для МКЭ были изложены в монографиях В.В. Шайдурова [5] и М.А. Ольшанского [6].

Перспектива решать краевые задачи с оптимальными вычислительными усилиями выглядела очень заманчиво, но развитие многосеточных методов пошло по традиционному для 1980-х годов пути адаптации отдельных компонент алгоритма к решаемой задаче. Достаточно быстро удалось разработать высокоэффективные многосеточные алгоритмы для решения уравнения Пуассона на равномерных сетках, однако усложнение решаемых краевых задач (нелинейность, анизотропия, разрывность коэффициентов и т. д.) быстро превратили классические многосеточные методы (КММ) в труднообозримое семейство алгоритмов, практически не поддающееся формализации. Из-за отсутствия четких критериев оптимальности адаптации компонент КММ к конкретной задаче трудно судить о достижимости оптимальной скорости сходимости. Сразу были предприняты многочисленные попытки разработать универсальный вариант КММ для решения широкого класса прикладных задач путем последовательного улучшения отдельных компонент [7, 8].

Хотя до сих пор не прекращаются попытки разработать универсальный вариант КММ, стала очевидной необходимость поиска иной формы алгоритмизации основополагающей идеи Р.П. Федоренко для последующей реализации в автономных программах. Универсальная Многосеточная Технология (УМТ) определяется как наиболее рациональная организация вычислений при решении краевых задач без контроля хода вычислительного процесса пользователем автономной программы.

1. Универсальная многосеточная технология

Для построения универсального алгоритма представляется перспективной взаимная адаптация исходной дифференциальной задачи, вычислительной сетки, способа аппроксимации, итерационных методов решения СЛАУ и архитектуры компьютера. Поэтому УМТ² состоит из двух частей: аналитической (адаптация краевой задачи к численным методам) и вычислительной (аппроксимация адаптированной краевой задачи и решение результирующей СЛАУ при помощи оригинального многосеточного метода).

²Термин «технология» используется в традиционном смысле, а именно как «совокупность приемов обработки», указывая на наличие аналитической и вычислительной частей.

1.1. Аналитическая часть технологии. Для наглядности рассмотрим краевую задачу вида

$$\frac{d}{dx} \left(k(x) \frac{du}{dx} \right) - q(x) u(x) = -f(x), \quad (1)$$

$$0 < x < 1, \quad u(0) = u_0, \quad u(1) = u_1, \quad k(x) \geq \alpha > 0, \quad q(x) \geq 0$$

для иллюстрации основных компонент УМГ. Аналитическая часть УМГ для краевых задач типа (1) состоит в представлении искомого решения $u(x)$ в виде суммы двух функций $c(x)$ и $\hat{u}(x)$, то есть

$$u(x) = c(x) + \hat{u}(x). \quad (2)$$

В последующих многосеточных итерациях³ сеточный аналог функции $\hat{u}(x)$ будет служить приближением к решению разностной краевой задачи, а сеточный аналог функции $c(x)$ — поправкой, вычисляемой на грубых сетках. Представление (2), называемое Σ -модификацией решения $u(x)$, является одной из форм адаптации решаемых краевых задач к УМГ.

Подстановка представления (2) в (1) приводит к следующей Σ -модифицированной форме исходной краевой задачи

$$\frac{d}{dx} \left(k(x) \frac{dc}{dx} \right) - q(x) c(x) = r(x), \quad c(0) = u_0 - \hat{u}(0), \quad c(1) = u_1 - \hat{u}(1), \quad (3a)$$

где правая часть $r(x)$ имеет вид

$$r(x) = -\frac{d}{dx} \left(k(x) \frac{d\hat{u}}{dx} \right) + q(x) \hat{u}(x) - f(x). \quad (3b)$$

В (3) члены с поправкой $c(x)$ переносятся в левую часть, а остальные — в правую.

Σ -модификация решения, используемая в УМГ, похожа на представление решения, которое применяется в КММ, но между ними существует два принципиальных отличия:

- Σ -модификация в УМГ осуществляется *перед* дискретизацией исходной краевой задачи для более точной формулировки разностных задач на грубых сетках и возможности гибкого изменения типа и/или порядка аппроксимации;
- Σ -модификация не является единственным способом адаптации краевых задач к УМГ. Представление искомого решения в виде произведения двух функций (так называемая Π -модификация) может оказаться более предпочтительной для ряда нелинейных задач [9].

Σ - и Π -модификация являются частными случаями аддитивного и мультипликативного выделения особенностей, соответственно.

1.2. Вычислительная часть технологии. Вычислительная часть УМГ связана с компьютерным счетом и состоит из построения мелкой и грубых сеток, аппроксимации модифицированной краевой задачи интегро-интерполяционным методом и решения полученных сеточных уравнения при помощи унифицированного многосеточного метода.

³Данные названия функций \hat{u} (approximation to the solution) и c (coarse grid correction) пришлось позаимствовать из КММ для сохранения устоявшейся терминологии.

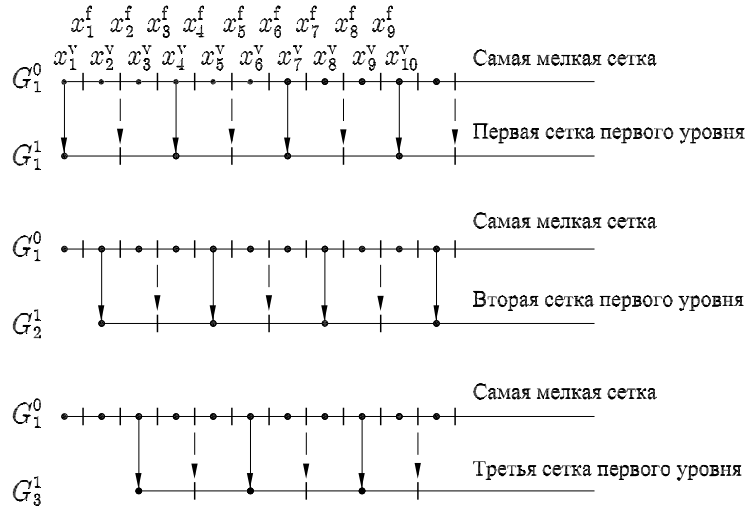


Рис. 1. Построение грубых сеток в универсальной многосеточной технологии

1.2.1. Построение самой мелкой сетки. Первый этап вычислительной части УМГ состоит в построении самой мелкой сетки G_1^0 в области $\Omega = [0, 1]$ для последующей аппроксимации модифицированной краевой задачи интегро-интерполяционным методом. Сетка G_1^0 состоит из двух множеств точек $G^v(0;1)$ и $G^f(0;1)$, которые задаются следующими соотношениями:

$$G^v(0;1) = \{x_i^v \mid x_i^v = h(i-1), \quad i = 1, 2, \dots, H_x^0 + 1, \quad h = (H_x^0)^{-1}\},$$

$$G^f(0;1) = \{x_i^f \mid x_i^f = 0.5(x_i^v + x_{i+1}^v), \quad i = 1, 2, \dots, H_x^0\}.$$

В общем случае до получения разностного аналога Σ -модифицированной краевой задачи будем называть x_i^v и x_i^f точками сетки. В КММ построение грубых сеток зависит от способа аппроксимации и/или расположения контрольных объемов на самой мелкой сетке [1]. В УМГ такой зависимости нет, поэтому конфигурация контрольных объемов может быть задана после построения грубых сеток, но перед аппроксимацией модифицированной краевой задачи. Другими словами, искомая сеточная функция, которая является решением разностной краевой задачи, может быть определена как в точках x_i^v , так и в x_i^f , но на построение грубых сеток это не влияет.

1.2.2. Построение грубых сеток. Построение грубых сеток в УМГ осуществляется посредством удаления двух точек из множеств $G^v(0;1)$ и $G^f(0;1)$ самой мелкой сетки G_1^0 , как показано на рис. 1. Выделим первую точку x_1^v из множества $G^v(0;1)$, которая будет первой точкой x^v более грубой сетки G_1^1 . Далее пропустим две точки x_2^v и x_3^v из множества $G^v(0;1)$. Четвертая точка x_4^v будет второй точкой x^v более грубой сетки G_1^1 , седьмая точка x_7^v – третьей точкой x^v более грубой сетки G_1^1 и т. д. Точка x_2^f из множества $G^f(0;1)$, расположенная посередине между точками x_1^v и x_4^v из множества $G^v(0;1)$, будет первой точкой x^f более грубой сетки G_1^1 . Удаляя по две точки из множества $G^f(0;1)$, нетрудно получить остальные точки x^f грубой сетки G_1^1 .

Далее аналогично построим еще одну грубую сетку G_2^1 , но построение начнем с точек x_2^v и x_3^f самой мелкой сетки G_1^0 , то есть со сдвигом на одну точку (рис. 1). Наконец, третья грубая сетка G_3^1 строится тем же самым образом, но начиная с

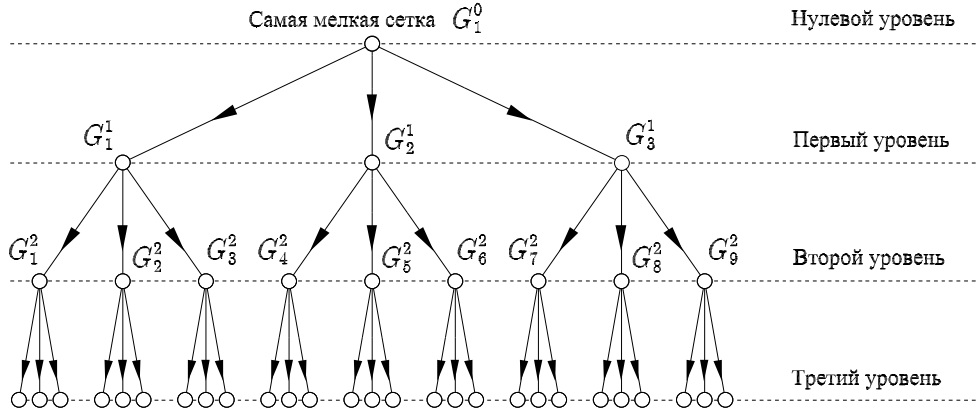


Рис. 2. Многосеточная структура

точек x_3^v и x_4^f самой мелкой сетки G_1^0 . Непосредственно из рис. 1 следуют основные свойства грубых сеток в УМТ:

- 1) грубые сетки G_1^1 , G_2^1 и G_3^1 не имеют общих точек, то есть

$$G_n^1 \cap G_m^1 = \emptyset, \quad n \neq m;$$

- 2) мелкая сетка G_1^0 представима в виде объединения грубых сеток G_1^1 , G_2^1 и G_3^1 , то есть

$$G_1^0 = \bigcup_{k=1}^3 G_k^1;$$

- 3) все сетки геометрически подобны, однако шаг грубых сеток G_1^1 , G_2^1 и G_3^1 в три раза больше, чем шаг сетки G_1^0 ;

- 4) вне зависимости от способа определения сеточных функций на самой мелкой сетке каждый контрольный объем на сетках G_1^1 , G_2^1 и G_3^1 является объединением трех контрольных объемов на сетке G_1^0 .

Самая мелкая сетка G_1^0 образует нулевой сеточный уровень, а три грубые сетки G_1^1 , G_2^1 и G_3^1 – первый сеточный уровень. Далее построение еще более грубых сеток осуществляется рекуррентным образом: каждая сетка G_i^L , $i = 1, \dots, 3^L$ уровня L рассматривается как самая мелкая сетка для трех грубых сеток G_j^{L+1} , $j = 1, \dots, 3^{L+1}$ следующего уровня $L + 1$. Девять еще более грубых сеток, полученных из трех сеток первого уровня, образуют второй сеточный уровень и так далее, как показано на рис. 2. Построение грубых сеток завершается, когда на грубых сетках останется всего несколько точек x^v и x^f . В дальнейшем совокупность самой мелкой и всех грубых сеток будет называться *многосеточной структурой*.

Номер сеточного уровня с самыми грубыми сетками, обозначаемый как L^+ , вычисляется перед построением грубых сеток. Положим, что большинство самых грубых сеток имеют три точки x^v или x^f . Тогда число точек самой мелкой сетки G_1^0 есть \bar{N} или $\approx 3^{L^++1}$. Отсюда L^+ определяется как

$$\bar{N} = H_x^0 + 1 \approx 3^{L^++1} \Rightarrow L^+ = \left\lfloor \frac{\lg \bar{N}}{\lg 3} - 1 \right\rfloor,$$

где квадратные скобки означают целую часть.

Сеточный уровень L состоит из сеток G_k^L , $k = 1, \dots, 3^L$ с шагом $h3^L$, где h есть шаг самой мелкой сетки G_1^0 . Сетки G_k^L состоят из двух множеств точек $G^v(L; k)$ и $G^f(L; k)$, которые могут быть как узлами, так и гранями контрольных объемов. Для удобства работы с многосеточной структурой обозначим *отображение индексов* точек данной сетки G_k^L на индексы точек самой мелкой сетки G_1^0 как $x_{\{i\}}^v$ и $x_{\{i\}}^f$, где $\{i\}$ и i есть индексы точек самой мелкой G_1^0 и данной сетки G_k^L соответственно. Например, отображение точек из множества $G^v(1; 1)$ сетки G_1^1 , показанной на рис. 1, есть $x_{\{1\}}^v = x_1^v$, $x_{\{2\}}^v = x_4^v$, $x_{\{3\}}^v = x_7^v$, ... Отображение индексов позволяет формулировать разностные краевые задачи на многосеточной структуре практически так же, как и в односеточных алгоритмах.

Поскольку N -мерная сетка представима в виде произведения N одномерных сеток, то построение грубых сеток осуществляется независимо в каждом пространственном направлении. В этом случае сеточный уровень L состоит из 3^{NL} сеток.

Перечислим основные свойства грубых сеток УМТ, которые будут использованы для численного решения краевых задач.

Свойство 1: каждый контрольный объем на сетках G_k^L представим в виде объединения 3^L контрольных объемов на самой мелкой сетке G_1^0 . В сочетании со свойством аддитивности определенного интеграла относительно подобластей это позволит существенно расширить класс краевых задач, решаемых унифицированным образом.

Свойство 2: каждая сетка G_k^L ($L \neq L^+$) представима в виде объединения трех соответствующих ей грубых сеток, что позволит исключить интерполяцию из УМТ. Как следствие, самая мелкая сетка G_1^0 представима в виде объединения всех сеток одного уровня:

$$G_1^0 = \bigcup_{k=1}^{3^L} G_k^L, \quad L = 0, \dots, L^+.$$

Свойство 3: сетки одного уровня не имеют общих точек, то есть

$$G_n^L \cap G_m^L = \emptyset, \quad n \neq m, \quad L = 1, \dots, L^+,$$

что позволит эффективно распараллеливать вычисления и экономно использовать память компьютера.

1.2.3. Аппроксимация модифицированных краевых задач на многосеточной структуре. Аппроксимация на многосеточной структуре интегро-интерполяционным методом мало отличается от «односеточной» аппроксимации [10, 11].

Поскольку на границах области $\Omega = [0, 1]$ заданы условия Дирихле, то точки сетки $x_{\{i\}}^v$ будут узлами, $x_{\{i\}}^f$ – гранями контрольных объемов, а сам объем определяется как

$$V_i = \{x \mid x_{\{i-1\}}^f \leq x \leq x_{\{i\}}^f\}.$$

Интегрирование уравнения (3а) по контрольному объему V_i приводит к следующему соотношению

$$-\frac{w(x_{\{i\}}^f) - w(x_{\{i-1\}}^f)}{h3^L} - \frac{1}{h3^L} \int_{x_{\{i-1\}}^f}^{x_{\{i\}}^f} q(x) c(x) dx = \frac{1}{h3^L} \int_{x_{\{i-1\}}^f}^{x_{\{i\}}^f} r(x) dx,$$

где $w = -kc'$. Использование простейших интерполяций приводит к следующей разностной схеме

$$\langle k \rangle_{\{i+1\}} \frac{c_{\{i+1\}} - c_{\{i\}}}{h^2 3^{2L}} - \langle k \rangle_{\{i\}} \frac{c_{\{i\}} - c_{\{i-1\}}}{h^2 3^{2L}} - \langle q \rangle_{\{i\}} c_{\{i\}} = \langle r \rangle_{\{i\}},$$

где

$$\langle q \rangle_{\{i\}} = \frac{1}{h3^L} \int_{x_{\{i-1\}}^f}^{x_{\{i\}}^f} q(x) dx$$

есть среднее значение функции $q(x)$ на контрольном объеме V_i и

$$\frac{1}{\langle k \rangle_{\{i\}}} = \frac{1}{h3^L} \int_{x_{\{i-1\}}^v}^{x_{\{i\}}^v} \frac{dx}{k(x)} \Rightarrow \langle k \rangle_{\{i\}} = \left(\frac{1}{h3^L} \int_{x_{\{i-1\}}^v}^{x_{\{i\}}^v} \frac{dx}{k(x)} \right)^{-1}.$$

Правая часть разностной Σ -модифицированной задачи

$$\langle r \rangle_{\{i\}} = \frac{1}{h3^L} \int_{x_{\{i-1\}}^f}^{x_{\{i\}}^f} r(x) dx$$

является среднеарифметическим значением невязок, вычисленных в тех узлах самой мелкой сетки, которые расположены внутри данного контрольного объема.

Первое отличие аппроксимации модифицированных краевых задач на многосеточной структуре от односеточной аппроксимации состоит в вычислении коэффициентов $\langle q \rangle_{\{i\}}$, $\langle k \rangle_{\{i\}}$ и правой части $\langle r \rangle_{\{i\}}$. Алгоритм быстрого вычисления интегралов на многосеточной структуре изложен в [12]. Второе отличие состоит в аппроксимации граничных условий на грубых сетках, чьи граничные узлы не совпадают с границами области. Аппроксимация условий Дирихле и Неймана подробно изложена в [9, 12].

1.2.4. Многосеточные итерации. Многосеточная итерация УМТ схематично показана на рис. 3. В УМТ используется только простейший (пилообразный) цикл (sawtooth cycle), который является частным случаем V-цикла [1]. Вычисления начинаются с уровня L^+ , состоящего из самых грубых сеток. Сначала на каждой сетке уровня L^+ вычисляют коэффициенты $\langle q \rangle_{\{i\}}$, $\langle k \rangle_{\{i\}}$ и правую часть $\langle r \rangle_{\{i\}}$, а затем выполняют сглаживающие итерации по мере достижения критерия останова. Поскольку сетки уровня L^+ содержат лишь несколько узлов, то, как правило, требуется несколько сглаживающих итераций для вычисления поправки на данном уровне. Далее осуществляется переход к следующему уровню $L^+ - 1$ с более мелкими сетками. Данный переход представляет собой изменение отображения индексов и не вносит никаких погрешностей в поправку, вычисленную на сетках уровня L^+ . На уровне $L^+ - 1$ вычисления проводятся аналогичным образом: сначала определяются коэффициенты $\langle q \rangle_{\{i\}}$, $\langle k \rangle_{\{i\}}$ и правая часть $\langle r \rangle_{\{i\}}$, а затем выполняются сглаживающие итерации. После выполнения сглаживающих итераций на самой мелкой сетке осуществляется пересчет приближения к решению ($\hat{u} := \hat{u} + c$) и обнуление поправки ($c := 0$). На этом многосеточная итерация заканчивается, проверяется критерий останова на самой мелкой сетке и в случае необходимости осуществляется следующая многосеточная итерация, начиная с уровня L^+ (рис. 3).

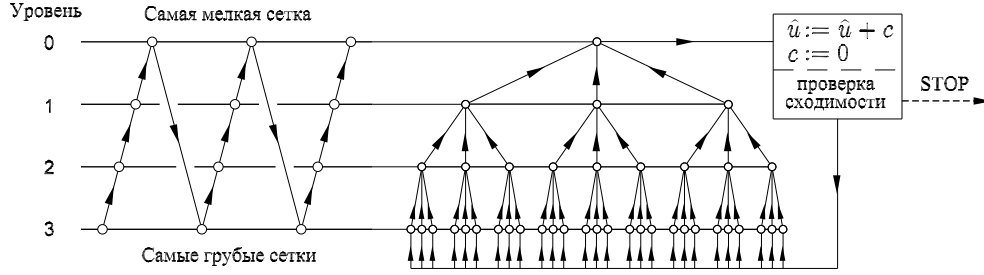


Рис. 3. Пилообразный цикл и многосеточная итерация

В многосеточных итерациях УМТ отсутствует предварительное сглаживание, чтобы избежать дополнительных (по отношению к односеточным алгоритмам) проблем при решении нелинейных задач.

2. Вычислительные эксперименты

Результаты вычислительных экспериментов, связанных с решением различных двух- и трехмерных краевых задач, приводятся в [9]. В данной статье будет показана лишь возможность гибкого изменения способа и/или порядка аппроксимации.

2.1. Изменения способа и/или порядка аппроксимации краевых задач. В качестве модельной рассмотрим задачу Дирихле для уравнения Пуассона

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -f(x, y), \quad u|_{\partial\Omega} = 0, \quad (4)$$

где область Ω является единичным квадратом.

Для наглядности рассмотрим наименее эффективный вариант УМТ, в котором множество используемых итерационных методов \mathfrak{M} состоит только из одного элемента, а именно метода Зейделя с точечным упорядочиванием неизвестных.

Краевая задача (4) записывается в следующей Σ -модифицированной форме:

$$\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} = r(x, y), \quad c|_{\partial\Omega} = -\hat{u}, \quad (5a)$$

где

$$r(x, y) = -f(x, y) - \frac{\partial^2 \hat{u}}{\partial x^2} - \frac{\partial^2 \hat{u}}{\partial y^2}. \quad (5b)$$

Интегрирование модифицированной задачи (5) по контрольному объему приводит к следующей разностной схеме:

$$\frac{c_{\{i-1j\}} - 2c_{\{ij\}} + c_{\{i+1j\}}}{h^2 3^{2L}} + \frac{c_{\{ij-1\}} - 2c_{\{ij\}} + c_{\{ij+1\}}}{h^2 3^{2L}} = \langle r \rangle_{\{ij\}}, \quad (6a)$$

где

$$\langle r \rangle_{\{ij\}} = \frac{1}{h^2 3^{2L}} \int_{x_{\{i-1\}}^f}^{x_{\{i\}}^f} \int_{y_{\{j-1\}}^f}^{y_{\{j\}}^f} \left(-f(x, y) - \frac{\partial^2 \hat{u}}{\partial x^2} - \frac{\partial^2 \hat{u}}{\partial y^2} \right) dy dx. \quad (6b)$$

В первом тесте (**Тест 1а**) правая часть Σ -модифицированной задачи (6b) на самой мелкой сетке аппроксимируется на пятиточечном шаблоне:

$$\langle r \rangle_{ij} = -f_{ij} - \frac{\hat{u}_{i-1j} - 2\hat{u}_{ij} + \hat{u}_{i+1j}}{h^2} - \frac{\hat{u}_{ij-1} - 2\hat{u}_{ij} + \hat{u}_{ij+1}}{h^2} + O(h^2).$$

Во втором тесте (**Тест 1б**) правая часть Σ -модифицированной задачи (6b) на самой мелкой сетке аппроксимируется на девятиточечном шаблоне:

$$\begin{aligned} \langle r \rangle_{ij} = & -f_{ij} + \frac{h^2}{12} \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right)_{ij} - \frac{1}{6h^2} (\hat{u}_{i-1j+1} + 4\hat{u}_{ij+1} + \hat{u}_{i+1j+1} + \\ & + 4\hat{u}_{i-1j} - 20\hat{u}_{ij} + 4\hat{u}_{i+1j} + \hat{u}_{i-1j-1} + 4\hat{u}_{ij-1} + \hat{u}_{i+1j-1}) + O(h^4). \end{aligned}$$

Предположим, что точное решение (u_e) рассматриваемой краевой задачи (4) имеет вид

$$u_e(x, y) = Q(x)Q(y), \quad Q(\varrho) = 20(e^\varrho + (1 - e)\varrho - 1). \quad (7)$$

Подстановка точного решения (7) в исходную задачу (4) определяет правую часть $f(x, y)$.

Скорость сходимости УМТ и точность численного решения будем оценивать при помощи следующих параметров:

а) относительной евклидовой нормы невязки

$$\mathcal{R}^{(q)} = \frac{\|A\hat{\mathbf{u}}^{(q)} - \mathbf{b}\|}{\|\mathbf{b}\|};$$

б) погрешности численного решения

$$\mathcal{E}^{(q)} = \max_{ij} |u_e(x_i^v, y_j^v) - \hat{u}_{ij}^{(q)}|,$$

где q – номер многосеточной итерации.

Во всех вычислительных экспериментах итерационное решение разностных краевых задач будем начинать с нулевого начального приближения, то есть

$$\hat{u}_{ij}^{(0)} = 0 \Rightarrow \mathcal{R}^{(0)} = 1 \quad \text{и} \quad \mathcal{E}^{(0)} = \max_{ij} |u_e(x_i^v, y_j^v)|.$$

Численное решение данной модельной краевой задачи осуществлялось на шестиуровневой многосеточной структуре ($L^+ = 5$) с самой мелкой сеткой 501×501 ($h = 1/500$). Сглаживающая итерация состояла из четырех итераций метода Зейделя с различными точечными упорядочиваниями неизвестных. Две сглаживающие итерации выполнены на каждом сеточном уровне. Результаты вычислительного эксперимента сведены в табл. 1.

Вычислительный эксперимент продемонстрировал важное преимущество УМТ по сравнению с КММ, которое возникает при Σ -модификации краевой задачи: для получения высокоточного решения достаточно повысить порядок аппроксимации правой части Σ -модифицированной задачи. Данный подход можно развить дальше, то есть для аппроксимации левой и правой частей Σ -модифицированной задачи использовать различные методы и различные сетки. Таким образом, УМТ можно обобщить для решения краевых задач на неструктурированных сетках.

Проиллюстрируем основные положения данного обобщения на примере решения задачи Дирихле в единичном квадрате. Построим триангуляцию для аппроксимации правой части Σ -модифицированной краевой задачи и вспомогательную структурированную сетку (ВСС) для аппроксимации левой части Σ -модифицированной краевой задачи (рис. 4).

Табл. 1

Сходимость технологии в Тесте 1

Тест 1a			Тест 1б		
q	\mathcal{R}	\mathcal{E}	q	\mathcal{R}	\mathcal{E}
0	1	17.95	0	1	17.95
1	$8.653 \cdot 10^{-3}$	$1.753 \cdot 10^{-1}$	1	$8.656 \cdot 10^{-3}$	$1.753 \cdot 10^{-1}$
2	$5.690 \cdot 10^{-5}$	$1.126 \cdot 10^{-3}$	2	$5.696 \cdot 10^{-5}$	$1.122 \cdot 10^{-3}$
3	$4.571 \cdot 10^{-7}$	$1.414 \cdot 10^{-5}$	3	$4.576 \cdot 10^{-7}$	$8.181 \cdot 10^{-6}$
4	$5.671 \cdot 10^{-9}$	$6.081 \cdot 10^{-6}$	4	$5.677 \cdot 10^{-9}$	$9.866 \cdot 10^{-8}$
5	$8.342 \cdot 10^{-11}$	$5.985 \cdot 10^{-6}$	5	$1.049 \cdot 10^{-10}$	$1.388 \cdot 10^{-10}$
6	$3.288 \cdot 10^{-12}$	$5.985 \cdot 10^{-6}$	6	$1.901 \cdot 10^{-11}$	$2.419 \cdot 10^{-11}$
7	$2.468 \cdot 10^{-12}$	$5.985 \cdot 10^{-6}$	7	$5.798 \cdot 10^{-12}$	$3.778 \cdot 10^{-11}$

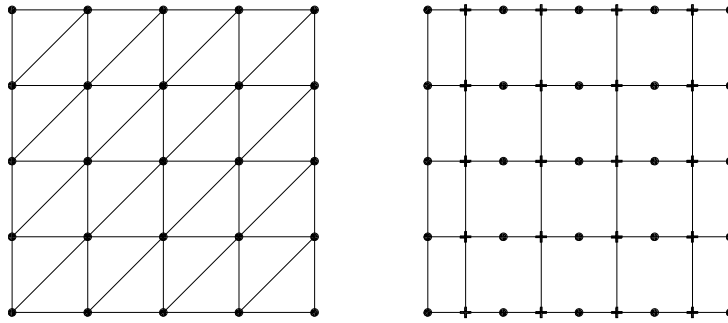


Рис. 4. Исходная триангуляция (слева) и BCC (справа)

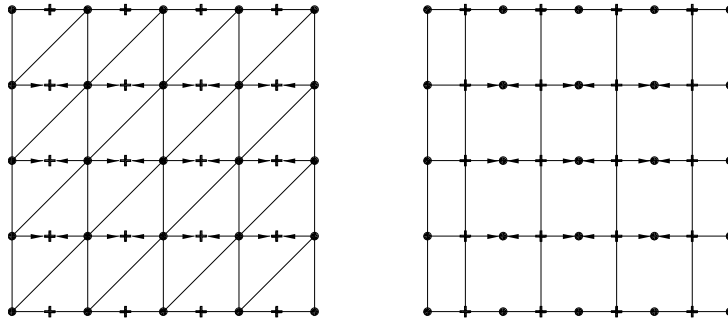


Рис. 5. Интерполяция невязки с триангуляции на BCC (слева) и интерполяция поправки с BCC на триангуляцию (справа)

Тогда многосеточная итерация УМТ представима в виде следующей последовательности действий:

1. Аппроксимация правой части Σ -модифицированной краевой задачи при помощи МКЭ на исходной триангуляции.
2. Вычисление невязки.
3. Интерполяция невязки с исходной триангуляции на BCC (рис. 5).
4. Аппроксимация левой части Σ -модифицированной краевой задачи при помощи интегро-интерполяционного метода на BCC.
5. Выполнение многосеточной итерации (рис. 3).

Табл. 2
Сходимость технологии в **Тесте 1в**

q	\mathcal{R}	\mathcal{E}
0	1	17.95
1	$1.536 \cdot 10^{-2}$	$2.045 \cdot 10^{-1}$
2	$8.566 \cdot 10^{-5}$	$1.956 \cdot 10^{-3}$
3	$6.123 \cdot 10^{-7}$	$1.841 \cdot 10^{-5}$
4	$7.252 \cdot 10^{-9}$	$8.401 \cdot 10^{-6}$
5	$9.663 \cdot 10^{-11}$	$6.005 \cdot 10^{-6}$
6	$4.446 \cdot 10^{-12}$	$5.985 \cdot 10^{-6}$
7	$3.474 \cdot 10^{-12}$	$5.985 \cdot 10^{-6}$

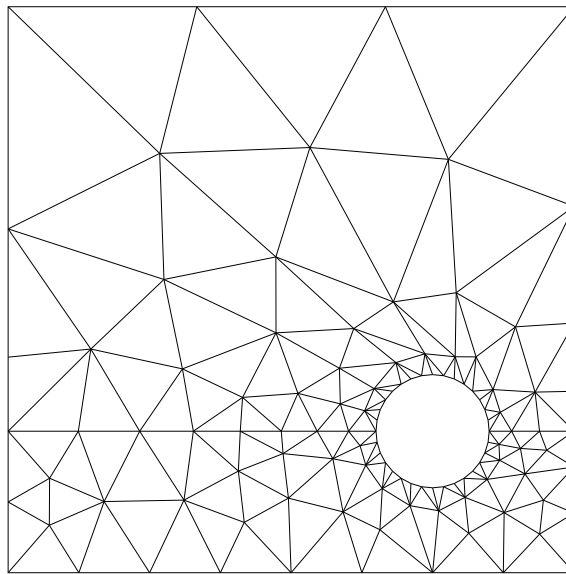


Рис. 6. Исходная триангуляция

6. Интерполяция поправки с ВСС на исходную триангуляцию (рис. 5).
7. Пересчет приближения к решению, обнуление поправки.
8. Проверка критерия останова, возврат к п. 1 (если необходимо).

Третий тест (**Тест 1в**) предназначен для иллюстрации сходимости УМТ для МКЭ и по своим основным условиям совпадает с первым тестом (**Тест 1а**). Результаты данного теста, сведенные в табл. 2, показывают что использование двух сеток для аппроксимации левой и правой части Σ -модифицированной краевой задачи не оказало заметного влияния на скорость сходимости УМТ. В отличие от алгебраических многосеточных методов, рассмотренный вариант УМТ может быть использован для решения нелинейных задач. По-видимому, при использовании неструктурированных сеток невозможно исключить интерполяцию из многосеточного алгоритма. Наличие такого проблемно-зависимого компонента, как интерполяция, неизбежно приведет к снижению уровня формализации при решении задач с разрывными коэффициентами. Отметим, что в УМТ интерполяция используется только на самой мелкой сетке.

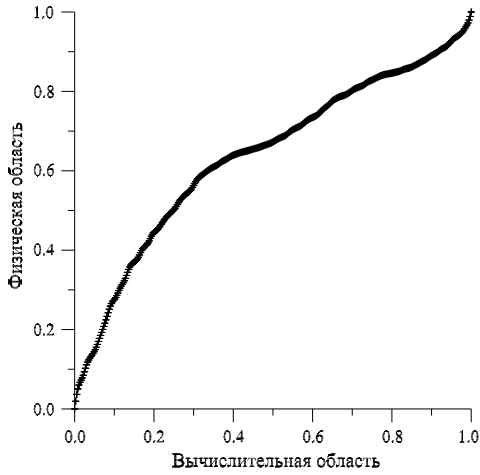


Рис. 7. Управляющая функция $\hat{\mathcal{F}} : \hat{x}_i \rightarrow x_i$

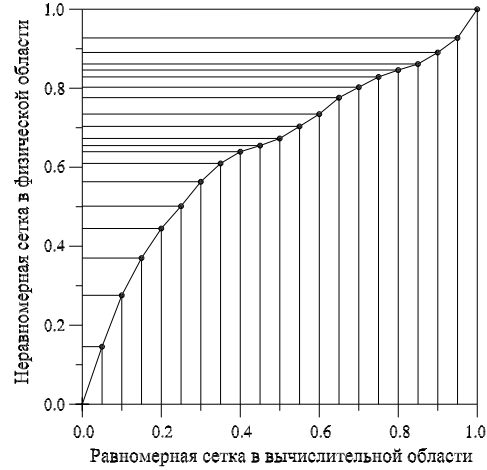


Рис. 8. Сетки в вычислительной и физической областях

Рассмотрим один из возможных вариантов построения ВСС. Предположим, что область вписана в единичный N -мерный куб. Пример двухмерной области показан на рис. 6. Разобьем область на треугольные элементы с вершинами $\Upsilon(x_k, y_k)$, $k = 1, 2, \dots, K$. Для области, показанной на рис. 6, $K = 409$. Составим упорядоченное по возрастанию множество абсцисс вершин, из которого удалены совпадающие элементы

$$\tilde{\Upsilon}^x = \{x_i \mid 0 = x_1 < x_2 < \dots < x_i < x_{i+1} < \dots < x_{\tilde{K}} = 1\}, \quad \tilde{K} \leq K.$$

В рассматриваемом примере $\tilde{K} = 368$. Далее построим на единичном отрезке равномерную сетку

$$\hat{\Upsilon}^x = \{\hat{x}_i \mid \hat{x}_i = \frac{i-1}{\tilde{K}-1}, i = 1, 2, \dots, \tilde{K}\}.$$

Функция $\hat{\mathcal{F}} : \hat{x}_i \rightarrow x_i$, обычно называемая управляющей, отображает равномерную сетку в вычислительной области на упорядоченное множество абсцисс вершин (рис. 7). Однако в данном случае число узлов ВСС будет приблизительно равным \tilde{K}^2 . Потребуем, чтобы число узлов ВСС было приблизительно равным числу вершин исходной триангуляции. Для этого построим еще одну равномерную сетку

$$\bar{\Upsilon}^x = \{\bar{x}_i \mid \bar{x}_i = \frac{i-1}{\bar{K}-1}, i = 1, 2, \dots, \bar{K}\},$$

где

$$\bar{K} = \left[\sqrt{\tilde{K}} \right] + 1,$$

причем квадратные скобки означают целую часть. В рассматриваемом примере $\bar{K} = 21$. Для отыскания положения узлов ВСС в физической области воспользуемся сплайн-интерполянтм управляющей функции $\hat{\mathcal{F}} : \hat{x}_i \rightarrow x_i$. На рис. 8 показана полученная равномерная сетка в вычислительной области, соответствующая ей сетка в физической области и сплайн-интерполянт управляющей функции $\bar{\mathcal{F}}$. Аналогичным образом строится сетка по каждому пространственному направлению. В данном примере число узлов ВСС составит $\bar{K}^2 = 441$, что несколько больше числа вершин ($K = 409$). Построенная по данной методике ВСС показана на рис. 9.

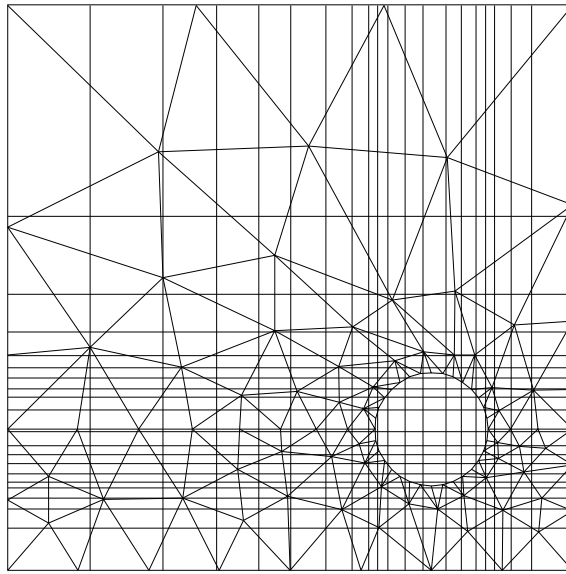


Рис. 9. Исходная триангуляция и ВСС

Уравнение вида

$$\frac{\partial}{\partial x} \left(\lambda^x \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda^y \frac{\partial u}{\partial y} \right) + \frac{\partial}{\partial z} \left(\lambda^z \frac{\partial u}{\partial z} \right) = -f(x, y, z)$$

в вычислительной области записывается как

$$\frac{\partial}{\partial \bar{x}} \left(\lambda^x \frac{\bar{x}'_x}{\bar{y}'_y \bar{z}'_z} \frac{\partial u}{\partial \bar{x}} \right) + \frac{\partial}{\partial \bar{y}} \left(\lambda^y \frac{\bar{y}'_y}{\bar{x}'_x \bar{z}'_z} \frac{\partial u}{\partial \bar{y}} \right) + \frac{\partial}{\partial \bar{z}} \left(\lambda^z \frac{\bar{z}'_z}{\bar{x}'_x \bar{y}'_y} \frac{\partial u}{\partial \bar{z}} \right) = -\frac{f(\bar{x}, \bar{y}, \bar{z})}{\bar{x}'_x \bar{y}'_y \bar{z}'_z},$$

то есть с точностью до вида коэффициентов и правой части совпадает с исходным уравнением. Производные \bar{x}'_x , \bar{y}'_y и \bar{z}'_z могут быть вычислены путем дифференцирования сплайн-интерполянтов соответствующих управляющих функций. В [9] показано, что в данном случае сглаживающие процедуры на основе метода Зейделя могут оказаться неэффективными. Поэтому на уровнях с грубыми сетками следует применять более сильные сглаживатели типа предобусловленных методов сопряженных градиентов. Пример построения комбинированной сглаживающей процедуры приведен в [9].

Summary

S.I. Martynenko. Formalization of Computations at Numerical Solution of Boundary Value Problems.

The paper represents a new robust multigrid technique for solving boundary value problems in black box manner. To overcome problem of robustness, the technique was based on incorporating the adaptation of boundary value problems to numerical methods, control volume discretisation and a new multigrid solver into a united computational algorithm. Special multiple coarse grid correction strategy makes it possible to obtain problem-independent transfer operators. As a result, the most of the modes are approximated on coarse grids in order to ensure the problem of smoothing on the finest grid to be the least demanding. Detailed description of the robust multigrid technique and examples of application for solving benchmark problems are given in the paper.

Key words: boundary value problem, multigrid methods, finite element method, numerical experiment.

Литература

1. *Wesseling P.* An Introduction to Multigrid Methods. – Chichester: John Wiley & Sons, 1991. – 284 p.
2. *Федоренко Р.П.* Релаксационный метод решения разностных эллиптических уравнений // Журн. вычисл. матем. и матем. физ. – 1961. – Т. 1, № 5. – С. 922–927.
3. *Федоренко Р.П.* Скорость сходимости одного итерационного метода // Журн. вычисл. матем. и матем. физ. – 1964. – Т. 4, № 3. – С. 227–235.
4. *Бажвалов Н.С.* О сходимости одного релаксационного метода при естественных ограничениях на эллиптический оператор // Журн. вычисл. матем. и матем. физ. – 1966. – Т. 6, № 5. – С. 861–883.
5. *Шайдулов В.В.* Многосеточные методы конечных элементов. – М.: Наука, 1989. – 288 с.
6. *Ольшанский М.А.* Лекции и упражнения по многосеточным методам. – М.: ФИЗМАЛИТ, 2005. – 168 с.
7. *Dendy Jr. J.E.* Black box multigrid // J. Comput. Phys. – 1982. – V. 48. – P. 366–386.
8. *Dendy Jr. J.E.* Black box multigrid for systems // Appl. Math. Comp. – 1986. – V. 19. – P. 57–74.
9. *Martynenko S.I.* Robust Multigrid Technique for black box software // Comp. Meth. in Appl. Math. – 2006. – V. 6, No 4. – P. 413–435.
10. *Калиткин Н.Н.* Численные методы. – М.: Наука, 1978. – 512 с.
11. *Самарский А.А.* Теория разностных схем. – М.: Наука, 1983. – 616 с.
12. *Мартыненко С.И.* Универсальная многосеточная технология для численного решения краевых задач на структурированных сетках // Вычислительные методы и программирование. – 2000. – Т. 1, Раздел 1. – С. 85–104.

Поступила в редакцию
25.06.07

Мартыненко Сергей Иванович – кандидат физико-математических наук, старший научный сотрудник отдела «Спецдвигатели и химмотология» ФГУП Центральный Институт Авиационного Моторостроения им. П.И. Баранова, г. Москва.

E-mail: *martyn_s@mail.ru*